1. Пусть в таблице users поля created_at и updated_at оказались незаполненными. Заполните их текущими датой и временем.

Заполняем таблицу users данными:

```
mysql> select * from users;
+----+-------------------------+-------------+---------------------+---------------------+
| id | name                    | birthday_at | created_at          | updated_at          |
+----+-------------------------+-------------+---------------------+---------------------+
|  1 | Вася Пупкин             | NULL        | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
|  2 | Петя Васин              | NULL        | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
|  3 | Антон Петров            | NULL        | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
|  4 | Вася Петин              | NULL        | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
+----+-------------------------+-------------+---------------------+---------------------+
```

Заполняем поля created_at и updated_at текущей датой и временем:

```
mysql> update users set created_at = now(), updated_at = now();
Query OK, 4 rows affected (0.00 sec)
Rows matched: 4  Changed: 4  Warnings: 0

mysql> select * from users;
+----+-------------------------+-------------+---------------------+---------------------+
| id | name                    | birthday_at | created_at          | updated_at          |
+----+-------------------------+-------------+---------------------+---------------------+
|  1 | Вася Пупкин             | NULL        | 2020-06-19 09:34:42 | 2020-06-19 09:34:42 |
|  2 | Петя Васин              | NULL        | 2020-06-19 09:34:42 | 2020-06-19 09:34:42 |
|  3 | Антон Петров            | NULL        | 2020-06-19 09:34:42 | 2020-06-19 09:34:42 |
|  4 | Вася Петин              | NULL        | 2020-06-19 09:34:42 | 2020-06-19 09:34:42 |
+----+-------------------------+-------------+---------------------+---------------------+
```

2. Таблица users была неудачно спроектирована. Записи created_at и updated_at были заданы типом VARCHAR и в них долгое время помещались значения в формате "20.10.2017 8:10". Необходимо преобразовать поля к типу DATETIME, сохранив введеные ранее значения.

Создаем неудачно спроектированную таблицу users:

```
mysql> CREATE TABLE users (
    ->   id INT UNSIGNED NOT NULL,
    ->   name VARCHAR(255) COMMENT 'Имя покупателя',
    ->   birthday_at DATE COMMENT 'Дата рождения',
    ->   created_at VARCHAR(100),
    ->   updated_at VARCHAR(100));
Query OK, 0 rows affected (0.01 sec)

mysql> describe users;
+-------------+------------------+------+-----+---------+-------+
| Field       | Type             | Null | Key | Default | Extra |
+-------------+------------------+------+-----+---------+-------+
| id          | int(10) unsigned | NO   |     | NULL    |       |
| name        | varchar(255)     | YES  |     | NULL    |       |
| birthday_at | date             | YES  |     | NULL    |       |
| created_at  | varchar(100)     | YES  |     | NULL    |       |
| updated_at  | varchar(100)     | YES  |     | NULL    |       |
+-------------+------------------+------+-----+---------+-------+
```

Заполняем данными строковом формате:

```
mysql> select * from users;
+----+----------+-------------+------------------+------------------+
| id | name     | birthday_at | created_at       | updated_at       |
+----+----------+-------------+------------------+------------------+
|  1 | Вася     | NULL        | 20.10.2017 8:10 | 20.10.2017 8:10 |
|  2 | Петя     | NULL        | 20.10.2018 8:10 | 20.10.2018 8:10 |
+----+----------+-------------+------------------+------------------+
```

Преобразуем строку в формат DATETIME с помощью функции STR_TO_DATE:

```
mysql> UPDATE users SET created_at = STR_TO_DATE(created_at, '%d.%m.%Y %k:%i'),
    ->   updated_at = STR_TO_DATE(updated_at, '%d.%m.%Y %k:%i');
Query OK, 2 rows affected (0.00 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql> select * from users;
+----+----------+-------------+---------------------+---------------------+
| id | name     | birthday_at | created_at          | updated_at          |
+----+----------+-------------+---------------------+---------------------+
|  1 | Вася     | NULL        | 2017-10-20 08:10:00 | 2017-10-20 08:10:00 |
|  2 | Петя     | NULL        | 2018-10-20 08:10:00 | 2018-10-20 08:10:00 |
+----+----------+-------------+---------------------+---------------------+
```

Изменяем тип столбцов на DATETIME:

```
mysql> ALTER TABLE users
    ->   CHANGE COLUMN `created_at` `created_at` DATETIME DEFAULT CURRENT_TIMESTAMP,
    ->   CHANGE COLUMN `updated_at` `updated_at` DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP;
Query OK, 2 rows affected (0.05 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> select * from users;
+----+----------+-------------+---------------------+---------------------+
| id | name     | birthday_at | created_at          | updated_at          |
+----+----------+-------------+---------------------+---------------------+
|  1 | Вася     | NULL        | 2017-10-20 08:10:00 | 2017-10-20 08:10:00 |
|  2 | Петя     | NULL        | 2018-10-20 08:10:00 | 2018-10-20 08:10:00 |
+----+----------+-------------+---------------------+---------------------+
```

3. В таблице складских запасов storehouses_products в поле value могут встречаться самые разные цифры: 0, если товар закончился и выше нуля, если на складе имеются запасы. Необходимо отсортировать записи таким образом, чтобы они выводились в порядке увеличения значения value. Нулевые запасы должны выводиться в конце, после всех записей.

Заполняем таблицу:

```
mysql> INSERT INTO
    ->     storehouses_products (storehouse_id, product_id, value)
    -> VALUES
    ->     (1, 1, 15),
    ->     (1, 1, 16),
    ->     (1, 2, 17),
    ->     (1, 4, 0),
    ->     (1, 3, 0);
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> select * from storehouses_products;
+----+--------------+------------+-------+---------------------+---------------------+
| id | storehouse_id | product_id | value | created_at          | updated_at          |
+----+--------------+------------+-------+---------------------+---------------------+
|  1 |            1 |          1 |    15 | 2020-06-19 10:49:52 | 2020-06-19 10:49:52 |
|  2 |            1 |          1 |    16 | 2020-06-19 10:49:52 | 2020-06-19 10:49:52 |
|  3 |            1 |          2 |    17 | 2020-06-19 10:49:52 | 2020-06-19 10:49:52 |
|  4 |            1 |          4 |     0 | 2020-06-19 10:49:52 | 2020-06-19 10:49:52 |
|  5 |            1 |          3 |     0 | 2020-06-19 10:49:52 | 2020-06-19 10:49:52 |
+----+--------------+------------+-------+---------------------+---------------------+
```

Выводим отсортированный список:

```
mysql> SELECT
    ->     value
    -> FROM
    ->   storehouses_products ORDER BY IF (value > 0, 0, 1), value;
+-------+
| value |
+-------+
|    15 |
|    16 |
|    17 |
|     0 |
|     0 |
+-------+
5 rows in set (0.00 sec)
```

4. (по желанию) Из таблицы users необходимо извлечь пользователей, родившихся в августе и мае. Месяцы заданы в виде списка английских названий ('may', 'august').

SELECT name FROM users WHERE DATE_FORMAT(birthday_at, '%M') IN ('may', 'august');

5. (по желанию) Из таблицы catalogs извлекаются записи при помощи запроса. SELECT * FROM catalogs WHERE id IN (5, 1, 2); Отсортируйте записи в порядке, заданном в списке IN.

SELECT * FROM catalogs WHERE id IN (3, 1, 2) ORDER BY FIELD(id, 3, 1, 2);